

LISTA ELEVILOR PARTICIPANȚI LA CERCUL DE INFORMATICĂ

Hai la olimpiada

Data: 14.01.2017

Tema: Backtracjing in plan

Formatori: Erhan Mihail

Backtracking în plan

În fișierul de intrare, labirint.in, se află scrise pe prima linie 2 numere, n si m, reprezentând dimensiunile unei matrici ce codifică un labirint. Pe următoarele n linii se află câte m termeni din mulțimea {0,1}, reprezentând imaginea codificată a unui labirint. Pe ultimile 2 linii se află scrise coordonatele punctului de start, respectiv a punctului de stop. Se cere afișarea în fisierul de ieșire, labirint.out, a tuturor soluțiilor prin care putem ajunge de la punctul de start la punctul de stop, precum și a celei mai scurte soluții.

Labirint.in

```
9 11
1 0 0 0 0 0 0 0 0 0 0
0 1 0 1 1 1 0 0 0 0 1
0 0 0 1 0 1 0 1 1 0 0
0 1 0 1 0 1 0 0 1 1 0
0 1 0 1 0 0 0 0 1 0 0
0 1 0 1 0 0 0 1 1 0 0
0 1 1 0 0 1 0 0 0 0 1
0 1 0 0 1 0 0 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0
4 4
2 6
```

Codul sursă

```
#include <fstream>
#include <iomanip>
using namespace std;
ifstream in("labirint.in");
ofstream out("labirint.out");
const int dx[]={-1,0,1,0};
const int dy[]={0,1,0,-1};
int a[50][50],xi,yi,xf,yf,n,m,minim,mim[50][50];
void citire(){
```

```

int i,j;
in>>n>>m;
minim=n*m;
for(i=0;i<n;i++)
    for(j=0;j<m;j++)
        {in>>a[i][j];
        }
in>>xi>>yi>>xf>>yf;
}
bool bune(int i,int j){
    if(i<0 || j<0 || i==n || j==n)return 0;
    return 1;
}
void afisare(){
    for(int i=0;i<n;i++){
        for(int j=0;j<m;j++){
            out<<setw(2)<<a[i][j]<<" ";
            out<<"\n";
        }
        if(a[xf][yf]<minim)
        {
            minim=a[xf][yf];
            for(int i=0;i<n;i++){
                {
                    for(int j=0;j<m;j++)mim[i][j]=a[i][j];
                }
            }
            out<<"\n";
        }
    }
}
void bktr(int i,int j,int pos){
    int ii,jj,k;
    for(k=0;k<4;k++){
        ii=i+dx[k];
        jj=j+dy[k];
        if(bune(ii,jj))
            if(a[ii][jj]==0)
            {
                a[ii][jj]=pos;
                if(ii==xf&&jj==yf)afisare();
                else bktr(ii,jj,pos+1);
                a[ii][jj]=0;
            }
    }
}
int main()
{
    citire();
    a[xi][yi]=1;

```

```

bktr(xi,yi,2);
out<<"Traseul minim:"<<minim<<"\n";
for(int i=0;i<n;i++){
    for(int j=0;j<m;j++){
        out<<setw(2)<<mim[i][j]<<" ";
        out<<"\n";
    }
    return 0;
}

```

Pentru exemplu dat problema are 98 de soluții.
Primele 3 soluții generate sunt:

```

1 0 0 0 0 0 0 0 0 0 0
0 1 0 1 1 1 0 0 0 0 1
0 0 0 1 0 1 5 1 1 0 0
0 1 0 1 0 1 4 0 1 1 0
0 1 0 1 1 2 3 0 1 0 0
0 1 0 1 0 0 0 1 1 0 0
0 1 1 0 0 1 0 0 0 0 1
0 1 0 0 1 0 0 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0

```

```

1 0 0 0 0 0 0 0 0 0 0
0 1 0 1 1 1 0 0 0 0 1
0 0 0 1 0 1 7 1 1 0 0
0 1 0 1 0 1 6 5 1 1 0
0 1 0 1 1 2 3 4 1 0 0
0 1 0 1 0 0 0 1 1 0 0
0 1 1 0 0 1 0 0 0 0 1
0 1 0 0 1 0 0 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0

```

```

1 0 25 26 27 28 29 30 31 0 0
0 1 24 1 1 1 34 33 32 0 1
21 22 23 1 0 1 35 1 1 0 0
20 1 0 1 0 1 0 0 1 1 0
19 1 0 1 1 2 3 0 1 0 0
18 1 0 1 0 0 4 1 1 0 0
17 1 1 0 0 1 5 0 0 0 1
16 1 12 11 1 0 6 1 1 1 1
15 14 13 10 9 8 7 0 0 0 0

```

Soluția optima este:

Traseul minim:5

```

1 0 0 0 0 0 0 0 0 0 0
0 1 0 1 1 1 0 0 0 0 1

```

```
0 0 0 1 0 1 5 1 1 0 0
0 1 0 1 0 1 4 0 1 1 0
0 1 0 1 1 2 3 0 1 0 0
0 1 0 1 0 0 0 1 1 0 0
0 1 1 0 0 1 0 0 0 0 1
0 1 0 0 1 0 0 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0
```

Temă acasă:

Se dă în fișierul de intrare, sub forma unei matrici, codificarea unui arhipelag de dimensiunile n linii și m coloane. Se cere să se determine din câte insule este format arhipelagul, precum și suprafața pe care o ocupă fiecare insula.

Insule.in

8 10

```
1 0 1 0 0 1 1 1 1 1
0 0 1 1 1 0 0 0 0 1
1 0 1 1 0 0 1 1 0 0
1 0 1 0 0 1 1 1 0 1
1 0 1 1 1 1 0 0 1 1
1 0 0 0 0 0 0 1 1 0
1 1 1 1 0 1 0 1 0 1
0 1 1 0 0 0 1 1 0 1
```